IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In Re Application of:	)	
	)	
GOLD, Stephen, et al.	)	Examiner: Dillon, S.A.
	)	
Serial No. 10/683,752	)	Group Art Unit: 2185
	)	
Filing Date: October 10, 2003	)	Conf. No.: 5052
	)	
For: PROVIDING A STATUS INDICATOR FOR A MEDIA JOB CATEGORY	)	Atty. Dkt.: 200309332-1
	)	

APPEAL BRIEF

Bruce E. Dahl, Esq.
Attorney for Appellants
PTO Registration No. 33,670
FENNEMORE CRAIG, P.C.
1700 Lincoln Street, Suite 2625
Denver, CO 80203
Telephone: (303) 291-3200

TABLE OF CONTENTS

TABLE OF AUTHORITIES

Cases: Page

Statutes and Regulations:

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In Re Application of:	)	
	)	
GOLD, Stephen, et al.	)	Examiner: Dillon, S.A.
	)	
Serial No. 10/683,752	)	Group Art Unit: 2185
	)	
Filing Date: October 10, 2003	)	Conf. No.: 5052
	)	
For: PROVIDING A STATUS INDICATOR FOR A MEDIA JOB CATEGORY	)	Atty. Dkt.: 200309332-1
	)	

APPEAL BRIEF

To: The Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This Appeal Brief is submitted in response to the final rejections of the claims dated February 26, 2007. A Notice of Appeal was filed on April 20, 2007.

REAL PARTY IN INTEREST

The real party in interest is Hewlett-Packard Development Company, LP, a limited partnership established under the laws of the State of Texas and having a principal place of business at 20555 S.H. 249 Houston, Texas 77070, USA (hereinafter "HPDC"). HPDC is a wholly owned affiliate of Hewlett-Packard Company, a Delaware Corporation, headquartered in Palo Alto, CA. The general or managing partner of HPDC is HPQ Holdings, LLC.

RELATED APPEALS AND INTERFERENCES

There are currently no related appeals or interferences known to Appellants, Appellants' legal representative, or the assignee which will directly affect, or be directly affected by, or have a bearing on, the Board's decision.

STATUS OF CLAIMS

Claims 1-20 are pending in the application. Claims 1-20 currently stand rejected. The rejections of claims 1-20 are appealed.

STATUS OF AMENDMENTS

No amendments have been filed subsequent to the issuance of the final office action.

SUMMARY OF CLAIMED SUBJECT MATTER

The invention as claimed is summarized below with reference to the independent claims. The claims contain reference numerals and reference to the specification and drawings. All references are shown in the application at least where indicated herein.

(Claim 1) A method (Figs. 1-8, ¶¶ 12-32) comprising:

receiving (205, Fig. 2, ¶¶ 15, 18, 20-23, 26, and 27) a service level objective for a media job category;

determining (210, Figure 2, ¶¶ 16-19) a status for each of a plurality of media jobs associated with the media job category; and

providing (215, Figure 2, Figure 4, ¶¶ 15, 18, 19, 24, 25, and 29-31) for a user a status indicator (602, 604, 606, Fig. 6 ¶¶ 29-31) for the media job category based on the service level objective and the status for each of the plurality of media jobs.

(Claim 12) A system (Figs. 1-8, ¶¶ 12-32) comprising:

a user interface (104, Fig. 1, ¶¶ 12, 14, 18, 20, and 32) to receive (205, Fig. 2, ¶¶ 15, 18, 20-23, 26, and 27) a service level objective for a media job category;

logic (102, Fig. 1, ¶¶ 12-14), communicatively coupled to said user interface (104), to determine (210, Figure 2, ¶¶ 16-19) a status for each of a plurality of media jobs associated with the media job category and to determine a status indicator (602, 604, 606, Fig. 6 ¶¶ 29-31) for the media job category based on the service level objective and the status for each of the plurality of media jobs; and

wherein said user interface (104) is further configured to provide the status indicator (602, 604, 606) for the media job category.

(Claim 17) At least one machine-readable medium having stored thereon sequences of instructions, which, when executed by a machine, cause the machine to perform the actions of:

receiving (205, Fig. 2, ¶¶ 15, 18, 20-23, 26, and 27) a service level objective for a media job category;

determining (210, Figure 2, ¶¶ 16-19) a status for each of a plurality of media jobs associated with the media job category;

providing (215, Figure 2, Figure 4, ¶¶ 15, 18, 19, 24, 25, and 29-31) a status indicator (602, 604, 606, Fig. 6 ¶¶ 29-31) for the media job category based on the service level objective and the status for each of the plurality of media jobs.

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Whether claims 1-20 are unpatentable under 35 U.S.C. §103(a) as being obvious over Legato *et al.*, "Legato NetWorker Administrator's Guide" (Legato) in view of Cox *et al.*, U.S. Patent No. 5,535,335 (Cox).

ARGUMENT

ISSUE: WHETHER CLAIMS 1-20 ARE UNPATENTABLE UNDER 35 U.S.C. §103(a) AS BEING OBVIOUS OVER LEGATO IN VIEW OF COX.

Legal Standard For Rejecting Claims
Under 35 U.S.C. §103

The test for obviousness under 35 U.S.C. §103 is whether the claimed invention would have been obvious to those skilled in the art in light of the knowledge made available by the reference or references. *In re Donovan*, 184 USPQ 414, 420, n. 3 (CCPA 1975). It requires consideration of the entirety of the disclosures of the references. *In re Rinehart*, 189 USPQ 143, 146 (CCPA 1976). All limitations of the claims must be considered. *In re Boe*, 184 USPQ 38, 40 (CCPA 1974). In making a determination as to obviousness, the references must be read without benefit of appellants' teachings. *In re Meng*, 181 USPQ 94, 97 (CCPA 1974). In addition, the propriety of a Section 103 rejection is to be determined by whether the reference teachings appear to be sufficient for one of ordinary skill in the relevant art having the references before him to make the proposed substitution, combination, or other modifications. *In re Lintner*, 173 USPQ 560, 562 (CCPA 1972).

A basic mandate inherent in Section 103 is that a piecemeal reconstruction of prior art patents shall not be the basis for a holding of obviousness. It is impermissible within the framework of Section 103 to pick and choose from any one reference only so much of it as will support a given position, to the exclusion of other parts necessary to the full appreciation of what such reference fairly suggests to one of ordinary skill in the art. *In re Kamm*, 172 USPQ 298, 301-302 (CCPA 1972). Put somewhat differently, the fact that the inventions of the references and of the appellants may be directed to concepts for solving the same problem does not serve as a basis for arbitrarily choosing elements from references to attempt to fashion appellants'

claimed invention. *In re Donovan, supra*, at 420.

In the case of *In re Wright*, 6 USPQ2d 1959 (Fed. Cir. 1988) (restricted on other grounds by *In re Dillon*, 16 USPQ2d 1897 (Fed. Cir. 1990), the Court of Appeals for the Federal Circuit decided that the Patent Office had improperly combined references which did not suggest the properties and results of the appellants' invention nor suggest the claimed combination as a solution to the problem which appellants' invention solved. The CAFC reached this conclusion after an analysis of the prior case law, at p. 1961:

> "We repeat the mandate of 35 U.S.C.§ 103: it is the invention as a whole that must be considered in obviousness determinations. The invention as a whole embraces the structure, its properties, and the problem it solves. See, e.g., *Cable Electric Products, Inc. v. Genmark, Inc.*, 770 F.2d 1015, 1025, 226 USPQ 881, 886 (Fed. Cir. 1985) ("In evaluating obviousness, the hypothetical person of ordinary skill in the pertinent art is presumed to have the 'ability to select and utilize knowledge from other arts reasonably pertinent to [the] particular problem' to which the invention is directed"), quoting *In re Angle*, 444 F.2d 1168, 1171-72, 170 USPQ 285, 287-88 (CCPA 1971); *In re Antonie*, 559 F.2d 618, 619, 195 USPQ 6, 8 (CCPA 1977) ("In delineating the invention as a whole, we look not only at the claim in question... but also to those properties of the subject matter which are inherent in the subject matter **and** are disclosed in the Specification") (emphasis in original).
>
> The determination of whether a novel structure is or is not "obvious" requires cognizance of the properties of that structure and the problem which it solves, viewed in light of the teachings of the prior art. See, e.g., *In re Rinehart*, 531 F.2d 1048, 1054, 189 USPQ 143, 149 (CCPA 1976) (the particular problem facing the inventor must be considered in determining obviousness); see also *Lindemann Maschinenfabrik GmbH v. American Hoist and Derrick Co.*, 730 F.2d 1452, 1462, 221 USPQ 481, 488 (Fed. Cir. 1984) (it is error to focus "solely on the product created, rather than on the obviousness or notoriousness of its creation") (quoting *General Motors Corp. v. U.S. Int'l Trade Comm'n*, 687 F.2d 476, 483, 215 USPQ 484, 489 (CCPA 1982), cert. denied, 459 U.S. 1105 (1983).
>
> Thus the question is whether what the inventor did would have been obvious to one of ordinary skill in the art attempting to solve the problem upon which the inventor was working. *Rinehart*, 531 F.2d at 1054, 189 USPQ at 149; see also *In re Benno*, 768 F.2d 1340, 1345, 226 USPQ 683, 687 (Fed. Cir. 1985) ("applicant's problem" and the prior art present different problems requiring different solutions")."

More recently, the CAFC has reiterated the necessity that motivation be identified in

choosing to combine prior art references for an obviousness type rejection. As stated by the Court of Appeals for the Federal Circuit in *In re Rouffet*, 47 USPQ2d 1453 (Fed. Cir. 1998) at 1457:

> "[V]irtually all [inventions] are combinations of old elements." *Environmental Designs, Ltd. V. Union Oil Co.*, 713 F.2d 693, 698, 218 USPQ 865, 870 (Fed.Cir. 1983)("Most, if not all, inventions are combinations and mostly of old elements."). Therefore an examiner may often find every element of a claimed invention in the prior art. If identification of each claimed element in the prior art were sufficient to negate patentability, very few patents would ever issue. Furthermore, rejecting patents solely by finding prior art corollaries for the claimed elements would permit an examiner to use the claimed invention itself as a blueprint for piecing together elements in the prior art to defeat the patentability of the claimed invention. Such an approach would be "an illogical and inappropriate process by which to determine patentability." *Sensonics, Inc. v. Aerosonic Corp.*, 81 F.3d 1566, 1570, 38 USPQ2d 1551, 1554 (Fed.Cir. 1996).
>
> To prevent the use of hindsight based on the invention to defeat patentability of the invention, this court requires the examiner to show a motivation to combine the references that create the case of obviousness. In other words, the examiner must show reasons that the skilled artisan, confronted with the same problems as the inventor and with no knowledge of the claimed invention, would select the elements from the cited prior art references for combination in the manner claimed."

A reference which teaches away from the appellants' invention may not properly be used in framing a 35 U.S.C. §103 rejection of appellants' claims. See *United States v. Adams*, 148 USPQ 429 (1966).

The Examiner's Rejections

The examiner rejected claims 1-20 under 35 U.S.C. §103(a) as being obvious over Legato and Cox for the reasons stated in the final office action. The examiner's rejections are improper in that the references fail to provide the suggestion or incentive required to combine them in the manner set forth by the examiner. Because of this lack of suggestion or incentive, neither reference can be used to establish the required *prima-facie* case of obviousness under Section

103. Moreover, even if the references were combined, the resulting combination would still fail to meet the limitations of the pending claims. Therefore, the obviousness rejections of claims 1-20 are improper and must be removed.

The present invention relates to the management of media used to backup data and claims methods and systems for providing a "status indicator" that indicates whether a prior media job (i.e., a backup job) met the service level objective for a media job category. Exemplary media job categories include one or more media movement categories, a device load category, a scratch media initialization category, and other categories of media jobs for managing media used to backup data. In one embodiment, a service level objective may be a desired percentage of media jobs that should be completed in a certain time. As illustrated in Figure 6 of the pending application, a user interface may be used to provide the status indicator for the media jobs in one or more of the media job categories (e.g., media movement, device load, and scratch initialization) based on the service level objective.

Stated another way, the "status indicator" of the present invention is indicative of the past performance of the media job (i.e., a backup process), that is, whether the media job met a desired or targeted performance level. If the media job met targeted performance level (i.e., service level objective), the status indicator will indicate as much. Likewise, the status indicator will also indicate when the service level objective was not met. The status indicator thus informs a user about whether a previous media job has met the required performance level. Simply stated, the "status indicator" provided by the present invention is thus *backward-looking and performance-based.*

The Legato reference is an administrator's guide for the Legato NetWorker (release 6.0, Unix version) software product. As described in the reference, the Legato NetWorker software product provides a solution for network storage management to help protect and manage data

across an entire network of computers. The NetWorker software product is described as managing data, creating a database of saved data, and making it easy to locate data for recovery. See, for example, page 37 of the Legato reference. While the Legato reference describes that the NetWorker software product "monitors data status," it does so only in the context of rotating "data through the tracking system as it reaches different levels of obsolescence." See page 38 of the Legato reference. However, and as confirmed by the examiner, the Legato reference fails to disclose several elements and limitations contained in the pending claims, including

> "receiving a service level objective for a media job category and determining a status indicator for the media job category based on the service level objective and the status or each of the plurality of media jobs; and wherein said user interface is further configured to provide the status indicator for the media job category." Final office action, page 6.

The Cox reference discloses a method and system for reporting the status of an aggregate resource in a network of interconnected real resources. Cox defines a real resource as

> "an operative piece of hardware or software residing within a network, for example a communications network arranged in some hierarchical or other interconnected topological arrangement whose interconnection scheme and last reported physical status are maintained in a network topological and status database by a network management system, such as IBM Corporation's NetView product." Cox, col. 1, line 65 - col. 2, line 5.

Cox then goes on to define an aggregate resource as "a logical resource composed of or containing a number of real resources." Cox, col. 2, lines 9-11. According to Cox, the status of the aggregate resource is calculated based upon the status of the real resources that it contains rather than being determined by its own characteristic "since a determination based upon the real status of underlying elements will prove more reliable in estimating the ability of the aggregate resource to function." Cox, col. 2, lines 13-16. Significantly, nowhere does Cox ever describe data backup operations, much less specific issues relating to the management of media used to backup data.

The examiner's obviousness rejections are erroneous in that nether Legato nor Cox provides the suggestion or incentive required to combine them in the manner urged by the examiner. While the NetWorker software disclosed by the Legato reference aids in network storage management, it fails to disclose several elements and limitations contained in each of the rejected claims, as noted above. While the examiner cites to various portions of the Cox reference as containing disclosures that would fill in the gaps (i.e., the missing elements and limitations), the problem remains that the examiner has failed to identify any suggestion or incentive in either Legato or Cox that would motivate a person having ordinary skill in the art to make the combination.

First, a person having ordinary skill in the art would view Legato as satisfactorily addressing the network storage problem, because it is a solution and is presumed to function well. Legato even provides a solution that "monitors data status and automatically rotates data through the tracking system as it reaches different levels of obsolescence." Legato mentions no need for additional status monitoring. Consequently, Legato cannot be said to provide the suggestion or incentive to provide any additional functionality in this regard. Of course, Cox makes no reference to data backup at all, much less identifies any problems or issues relating to the management of the backup process. The only place where there is identified a need to provide additional status of the backup process is in the teachings of the present invention. However, the teachings of the present invention cannot be used as a guide to piece together the prior art.

Notwithstanding the foregoing, even if it were proper to combine Legato and Cox (which is denied), the resulting combination would fail to meet the limitations of the pending claims. As stated above, the Cox reference is concerned with "the problem of maintaining an accurate awareness of the status of logical or aggregate resources in a communications network," (col. 1, lines 21-24) so that an estimation can be made of the ability of an aggregate resource to function

now and in the future. Stated another way, the "status" provided by Cox is *forward-looking and capacity-related*. That is, the "status" of Cox's "aggregate resource" relates to the current ability of the aggregate resource to perform its intended function and at what capacity (e.g., "satisfactory," "degraded," "severely degraded," "unsatisfactory," as illustrated in Figs 2B and 2C of Cox).

In contrast, the "status indicator" provided by the present invention is *backward-looking and performance-based*. That is, the "status indicator" provided by the present invention relates to the past performance of the media job, i.e., whether it met the "service level objective." An analogy is useful in understanding the fundamental differences between the present invention and the teachings of the Cox reference.

For example, an analogy to the present invention in an employment context would be an employee performance review. That is, the past performance of the employee is reviewed in the context of certain performance criteria (i.e., a "service level objective") in order to determine whether the employee met the performance criteria. The "status indicator" would then be the result of a comparison of the past performance of the employee with the performance criteria. That is, the "status indicator" in the context of the present invention is *backward-looking and performance-based*.

An analogy to the Cox reference in an employment context would involve a determination of the current status of an employee and the capacity of that employee to perform work in the future. It does not involve an assessment of the past performance of the employee or whether that performance met certain performance criteria. It asks only "is the employee available to do work (now and in the future), and at what capacity?" That is, the "status" reported by Cox (i.e., the aggregate resource status) is *forward-looking and capacity-related*.

Of course, the analogies are imperfect, but still serve to illustrate the fundamental

differences between the present invention and the Cox reference. Thus, even if Cox and Legato were combined, the resulting combination would fail to meet the limitations of the pending claims because of these fundamental differences between Cox and the present invention. That is, the resulting combination of Cox and Legato would result a system wherein the "status" would be forward-looking and capacity-related (like Cox). Moreover, the "status" would relate to aggregate resources, not to whether media jobs met any service level objective, because Cox provides no teachings in this regard. Certainly, the combination would not result in a system wherein the status would be backward-looking and performance-based (as inherently required by the pending claims), because Cox teaches away from a status that would be backward-looking and performance-based.

In the final office action, the examiner asserts that Cox's "aggregate resource" is the same as a "media job category." This is not a reasonable construction of either term. Cox's "aggregate resource" is defined as a logical resource composed of or containing a number of real resources. On the other hand, a "media job category" in the context of the pending claims may include "one or more media movement categories. . . a device load category, a scratch media initialization category, and other categories of media jobs for managing media used to backup data." See paragraph 12 of the pending application. Clearly, a "category" is not a "resource," nor can a media job category be equated to a "resource."

In summation, then, claim 1 is allowable because neither Legato nor Cox provide the suggestion or incentive required to combine them in the manner required by pending claim 1. In addition, even if it were proper to combine Legato and Cox, which is denied, the resulting combination still would fail to make obvious pending claim 1, because the "status" provided by Cox is forward-looking and capacity-related, whereas the "status indicator" of the pending claims is backward-looking and performance-based. Of course, Cox's "aggregate resource" is not a

"media job category," thus cannot possibly meet the limitations of claim 1 in that regard. Consequently, claim 1 is not *prima-facie* obvious over Legato and Cox.

Dependent claims 2-11 are allowable over Legato and Cox at least because they depend from claim 1, which is allowable over Legato and Cox.

Independent claim 12 is allowable over Legato and Cox because neither reference provides the suggestion or incentive required to combine them in the manner required by claim 12. Moreover, even if it were proper to combine Legato and Cox, the resulting combination still would not meet the limitations of pending claim 12, because the "status" provided by Cox is forward-looking and capacity-related, whereas the "status indicator" of the pending claims is backward-looking and performance-based. In addition, Cox's "aggregate resource" is not a "media job category," thus cannot meet the limitations of claim 12. Consequently, claim 12 is not *prima-facie* obvious over Legato and Cox.

Dependent claims 13-16 are allowable over Legato and Cox at least because they depend from claim 12, which is allowable over Legato and Cox.

Independent claim 17 is not obvious over Legato and Cox because neither reference provides the suggestion or incentive required to combine them in the manner required by claim 17. Even if Legato and Cox were combined, the resulting combination still would not meet the limitations of pending claim 17, because the "status" provided by Cox is forward-looking and capacity-related, whereas the "status indicator" of the pending claims is backward-looking and performance-based. Moreover, Cox's "aggregate resource" is not a "media job category." Therefore, claim 17 is not *prima-facie* obvious over Cox and Legato.

Dependent claims 18-20 are not obvious in light of Legato and Cox at least because they depend from claim 17, which is not obvious in light of Legato and Cox.

CONCLUSION

Claims 1-20 are allowable because neither Legato nor Cox provide the suggestion or incentive required to combine them in the manner required by pending claims. Moreover, even if it were proper to combine Legato and Cox, which is denied, the resulting combination still would fail to make obvious the pending claims, because the "status" provided by Cox is forward-looking and capacity-related, whereas the "status" of the pending claims is backward-looking and performance-based. That is, the "status" provided by Cox is not the same as the "status" in the pending claims. In addition, Cox's "aggregate resource" is not a "media job category," as used in the pending claims. Therefore, Appellants respectfully requests the Board to reverse the rejections of claims 1-20.

Respectfully submitted,

By: _____

Bruce E. Dahl, Esq.
Attorney for Appellants
PTO Registration No. 33,670
FENNEMORE CRAIG, P.C.
1700 Lincoln Street, Suite 2625
Denver, CO 80203
(303) 291-3200

Date: 6-20-07

CLAIMS APPENDIX

1. A method comprising:

receiving a service level objective for a media job category;

determining a status for each of a plurality of media jobs associated with the media job category; and

providing for a user a status indicator for the media job category based on the service level objective and the status for each of the plurality of media jobs.

2. The method of claim 1, wherein determining a status comprises determining whether each of the plurality of media jobs completed within a due time for each respective media job.

3. The method of claim 1, wherein receiving a service level objective comprises receiving a first desired percentage of media jobs to be completed within a due time associated with the media job.

4. The method of claim 3, wherein providing a status indicator comprises:

calculating a calculated percentage of the plurality of media jobs that were completed within the due time associated with the media job;

if the calculated percentage is less than the first desired percentage, providing a critical status indicator.

5. The method of claim 3, further comprising receiving a second desired percentage

of media jobs to be completed within the due time associated with the media job, the second desired percentage greater than the first desired percentage.

6. The method of claim 5, wherein providing a status indicator comprises:

calculating a calculated percentage of the plurality of media jobs that completed within the due time associated with the media job;

if the calculated percentage is less than the first desired percentage, providing a critical status indicator;

if the calculated percentage is greater than the first desired percentage and less than the second desired percentage, providing a warning status indicator; and

if the calculated percentage is greater than the second desired percentage, providing an OK status indicator.

7. The method of claim 1, further comprising:

receiving a second service level objective comprising a desired percentage of media to have a known location;

calculating a calculated percentage of media of a total number of media having a known location; and

providing a second status indicator for media having a known location based on the calculated percentage and the second service level objective.

8. The method of claim 1, wherein providing a status indicator comprises providing one of a critical status, a warning status, and an OK status.

9. The method of claim 1, wherein the media job category comprises one of a media movement category, a device load category, and a scratch media initialization category.

10. The method of claim 9, wherein the media movement category is one of a vaulting category for media jobs that move media having protected data to a different location, a scratch category for media jobs that move media having data that exceeded a protected time period to a scratch bin, and a check out request category for media check out jobs.

11. The method of claim 1, further comprising displaying one or more overdue media jobs associated with the media job category.

12. A system comprising:

a user interface to receive a service level objective for a media job category;

logic, communicatively coupled to said user interface, to determine a status for each of a plurality of media jobs associated with the media job category and to determine a status indicator for the media job category based on the service level objective and the status for each of the plurality of media jobs; and

wherein said user interface is further configured to provide the status indicator for the media job category.

13. The system of claim 12, further comprising:

media job information comprising a plurality of media jobs associated with the media job category, each of the plurality of media jobs having a due time; and

wherein said logic is to determine the status using said media job information.

14. The system of claim 13, wherein said media job information comprises information on at least one of a plurality of media movement jobs, a plurality of device load jobs, and a plurality of scratch media initialization jobs.

15. The system of claim 12, wherein said logic is further configured to calculate a calculated percentage of the plurality of media jobs that were completed within a due time associated with the media job.

16. The system of claim 12, wherein the service level objective comprises a desired percentage of media jobs to be completed within a due time associated with the media job and said logic is configured to determine a critical status indicator if the calculated percentage is less than the desired percentage.

17. At least one machine-readable medium having stored thereon sequences of instructions, which, when executed by a machine, cause the machine to perform the actions of:

receiving a service level objective for a media job category;

determining a status for each of a plurality of media jobs associated with the media job category;

providing a status indicator for the media job category based on the service level objective and the status for each of the plurality of media jobs.

18. The medium of claim 17, wherein the instructions for providing a status indicator comprise instructions which, when executed by the machine, cause the machine to perform the actions of:

calculating a calculated percentage of the plurality of media jobs that were completed within a due time for each respective media job;

if the calculated percentage is less than the service level objective, providing a critical status indicator, wherein the service level objective comprises a desired percentage of media jobs to be completed within the due time for each respective media job.

19. The medium of claim 17, further comprising instructions which, when executed by the machine, cause the machine to perform the actions of:

receiving a second service level objective comprising a desired percentage of media to have a known location; and

calculating a calculated percentage of media of a total number of media having a known location; and

providing a second status indicator for media having a known location based on the calculated percentage and the second service level objective.

20. The medium of claim 17, further comprising instructions which, when executed by the machine, cause the machine to perform the action of displaying one or more overdue media jobs associated with the media job category.

EVIDENCE APPENDIX

None.

RELATED PROCEEDINGS APPENDIX

None.

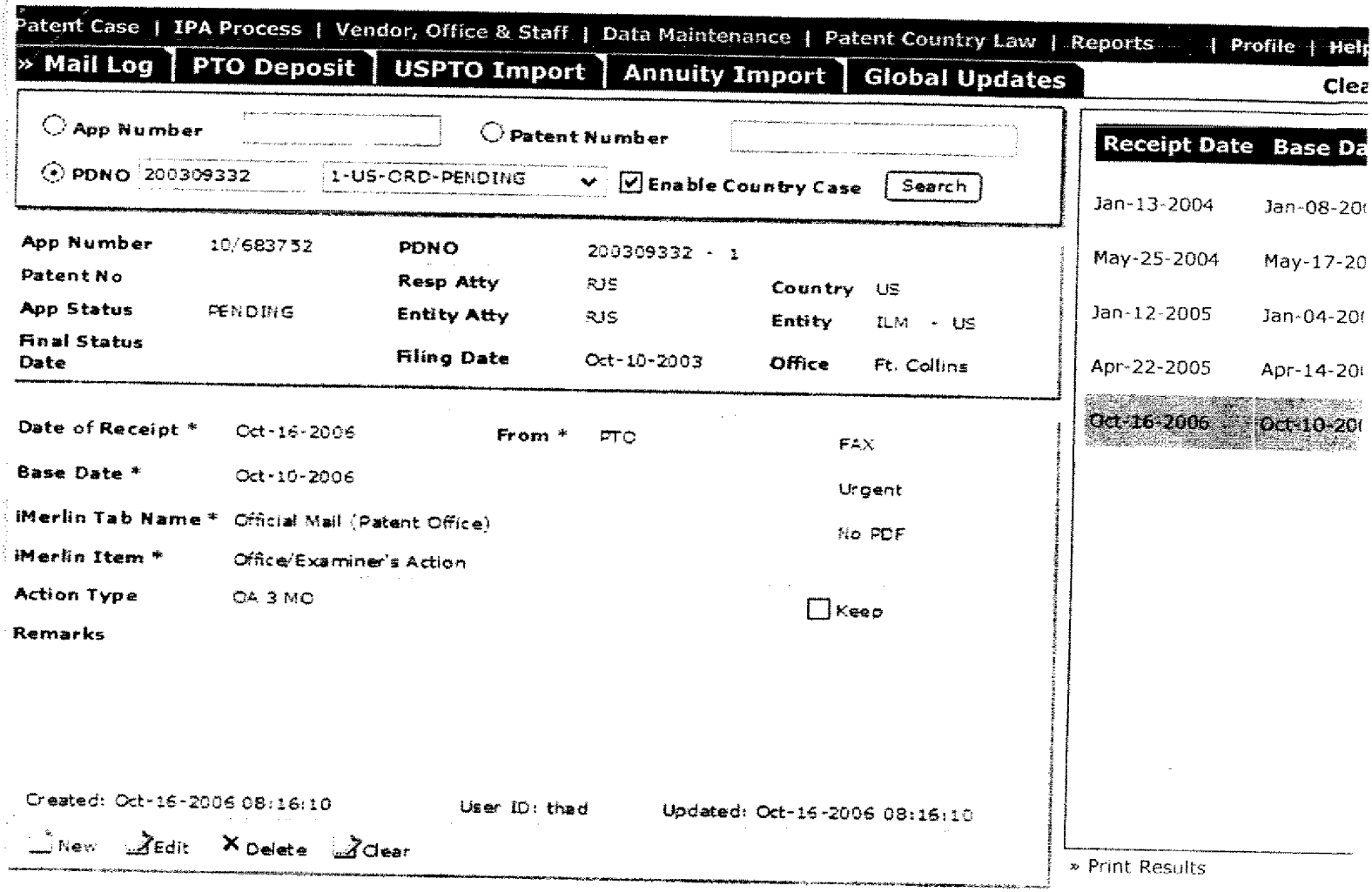
REFERENCE APPENDIX

References Relied on By Examiner in Final Office Action.

Copies of the following references are attached hereto for the Board's convenience:

1. Legato, *et al.*, "Legato NetWorker Administrator's Guide."

2. U.S. Patent No. 5,535,335, "Method and System for Reporting the Status of an Aggregate Resource Residing in a Network of Interconnected Real Resources," of Cox *et al.*


Patent Case | IPA Process | Vendor, Office & Staff | Data Maintenance | Patent Country Law | Reports | Profile | Help
» Mail Log
PTO Deposit
USPTO Import
Annuity Import
Global Updates
Clea
App Number
Patent Number
PDNO 200309332
1-US-ORD-PENDING
Enable Country Case
Search
Receipt Date
Base Da
Jan-13-2004
Jan-08-20
May-25-2004
May-17-20
Jan-12-2005
Jan-04-20
Apr-22-2005
Apr-14-20
Oct-16-2006
Oct-10-20
App Number 10/683732
PDNO 200309332 - 1
Patent No
Resp Atty RJS
Country US
App Status PENDING
Entity Atty RJS
Entity ILM - US
Final Status Date
Filing Date Oct-10-2003
Office Ft. Collins
Date of Receipt * Oct-16-2006
From * PTO
FAX
Base Date * Oct-10-2006
Urgent
iMerlin Tab Name * Official Mail (Patent Office)
No PDF
iMerlin Item * Office/Examiner's Action
Action Type OA 3 MO
Keep
Remarks
Created: Oct-16-2006 08:16:10
User ID: thad
Updated: Oct-16-2006 08:16:10
New
Edit
Delete
Clear
» Print Results

Legato NetWorker®

Administrator's Guide

Release 6.0
UNIX® Version

Legato NetWorker Administrator's Guide, Release 6.0, UNIX Version

August 2000

22-1948-01

Chapter 1: Introduction

This chapter introduces you to the NetWorker software product line. It explains how NetWorker products back up and recover data, and how you license NetWorker software for use. It also describes related NetWorker products.

NetWorker Storage Management

It is important to back up the data on computer systems to protect them from the loss of valuable data. In a networked environment, where users depend on shared data and the amount of data grows rapidly as systems are added to the network, the need to manage data becomes crucial.

Legato NetWorker products provide a solution for network storage management; it protects and helps manage data across an entire network of computers. The NetWorker software saves valuable administrator time by speeding up and simplifying daily operations of storage management. A graphical user interface (GUI) guides you through administering and configuring your network for storage management. As the NetWorker software manages your data, it creates a database of saved data, making it easy to locate data for recovery. Furthermore, as the network and amount of data grow, the NetWorker environment provides the capacity and performance to handle the load.

NetWorker software features include a storage management application that directs high performance to a wide range of storage devices. The NetWorker software product is made up of the following components:

- NetWorker client software, which is a collection of processes and programs installed on the computers that contain data to be managed.

- NetWorker server software, which is a collection of processes and programs installed on the computer that performs the NetWorker services, including data backup, recovery, archiving, and NetWorker Hierarchical Storage Management (HSM™).
- NetWorker storage node software, which is a collection of processes and programs installed on computers that control their attached storage devices during NetWorker operations, including backup, recovery, archive, and HSM.

You can configure your NetWorker software to back up data to storage devices that are directly connected to the NetWorker server, or you can create a NetWorker *storage node*, which is a separate computer with one or more storage devices connected. The NetWorker server maintains the resources and files that keep track of NetWorker data. The storage node computer controls the storage devices and media.

You can direct backup data, archive data, and migrate data to specific collections of volumes, according to particular data characteristics. As data is written to storage media, the NetWorker server creates a database to track the location of all managed data. The NetWorker server monitors data status and automatically rotates data through the tracking system as it reaches different levels of obsolescence, as defined by you.

You and your users can browse an index of recoverable files, create reports that describe the status of data or media, and recover data to a point in time specified by the user. In response to a data recovery request, the NetWorker storage management system locates the volume that contains the requested data and either directs a device to mount the appropriate volume for retrieval, or sends a message to the operator to request the volume by name. In this way, the NetWorker system manages all storage volume operations.

Cross-Platform Management

The NetWorker software is a cross-platform product for storage management. You can administer a NetWorker server from a workstation running UNIX, Windows®, or Novell NetWare®, if the workstation has the NetWorker client software installed and is connected by a network. Additionally, a NetWorker client on one platform can back up its data to a NetWorker server of a different platform.

The NetWorker server can direct and conduct administration services for any client or server on the network that has the NetWorker software installed and is recognized by the NetWorker server. The NetWorker administration program interface offers you only the options that are relevant to a particular NetWorker client, so you do not need to know the configuration of individual computers before you start a monitoring session.

Performance

NetWorker software has many standard and optional performance features:

- Parallelism, which allows several *save streams* to flow to the server or storage node at the same time.
- Multiplexing, which lets more than one save stream write to the same device at the same time.
- Client parallelism, which lets the client send more than one save stream at a time.
- Session management, which enables you to control the number of save streams per device to maximize the performance of each device.
- Backup to file devices and optional subsequent staging to near-line or offline volumes.
- Optional NetWorker software additions, such as:
  - NetWorker Autochanger Module
  - NetWorker Silo Software Module
  - NetWorker Archive™
  - NetWorker Modules
  - NetWorker SNMP (Simple Network Management Protocol)
  - Legato GEMS™ SmartMedia®
  - Remote Library Managers (RLM)
  - NetWorker HSM
  - HP-UX® Clustering
  - Solaris™ Clustering
  - Legato License Manager

Ease of Use

NetWorker software provides tools to make protection of critical data easy to manage. With the NetWorker software, you can:

- Use either the graphical NetWorker interface or the command line to manage NetWorker operations.
- Administer and configure NetWorker functions from any computer on the network.
- Grant permission to provide the capability for recovery of one client's data to another client computer of the same operating system.
- Obtain immediate answers to questions by accessing online help and man pages.
- Take advantage of the automatic media management feature to allow the NetWorker server or storage node to label and mount volumes as needed for backups.
- Use the Tech Dialog and technical bulletins on the Legato web site to find answers to common questions.

Scalability

You can add to your current NetWorker software purchase as your storage management needs grow. For example, you can:

- Upgrade the basic level of server functionality, add support for additional (or larger) autochangers, add support for more clients, or add optional software modules without the need to reinstall the server software.
- Add special NetWorker Module client software to back up databases and other nonfilesystem data.
- Add support for remote storage nodes to control backup devices, while the data management tasks remain centralized on a controlling NetWorker server.
- Introduce Legato GEMS into your enterprise environment to streamline storage management and provide comprehensive administrative policies that control media, devices, software, licensing, and NetWorker servers and clients. The Legato GEMS Java™-based interface allows you to administer, configure, and monitor your storage management applications from one location.

How the NetWorker Software Works

The *NetWorker server* is the computer on your network that runs the NetWorker server software, stores the client indexes, media database, and resource files; and provides backup and recovery services to the clients on the network. You can connect storage devices to the NetWorker server or attach them to a *storage node*. The *NetWorker client* must have the client software installed and must be included in the server's list of clients. All server and storage node computers that you want to back up a client to must be listed in the client's *nsrhost* file, which is created when you install the client software.

The NetWorker server backs up client data in increments called *save sets*. A save set typically comprises all the backed-up data from a single filesystem or logical disk. Save sets are saved to *volumes* (for example, tapes, magnetic disk, or optical disk) mounted in backup devices attached to the server. The server uses a *pool* to sort specific data to preselected volumes to help manage your data and collection of volumes.

The NetWorker server maintains records of the client save sets and the volumes in a *client file index* for each client and in a *media database* on the server. The NetWorker software uses these records to track the saved files and the volumes on which the files are stored. The client file index and media database contain the tracking information for all data controlled by the NetWorker server, whether the data is written to a device attached to the server or a device attached to a storage node.

When the NetWorker server backs up files, you might receive a request to mount a writable volume in the server's backup device. When a NetWorker user recovers files, you receive a request to mount a specific volume by its name. To fulfill either request, you only need to mount the requested volume or volumes in the device attached to the server. If you back up to an autochanger, the requested volume is automatically loaded if it is stored in the autochanger.

This guide uses the term *autochanger* to refer to a variety of robotic libraries, including jukebox, autoloader, carousel, library, near-line storage, and datawheel devices.

The NetWorker software supports many of the latest backup devices that provide the highest performance and most reliable solutions for your network backup and recovery requirements. Refer to the *Compatibility Guide*, available from *www.legato.com*, for the current list of supported devices.

NetWorker Resources and Attributes

NetWorker performance and functionality are configured through NetWorker *resources*. Resources are the building blocks the NetWorker software uses to license, configure, and operate NetWorker clients and servers to back up, store, and recover data. NetWorker resources are represented in the Administrator program using windows and icons.

Most resources can be created, edited, and deleted by administrators with the proper permissions. These resources contain the attributes for creating and editing resources. NetWorker *attributes* are individual components of a resource. Attributes are represented by the various text boxes, drop-down lists, and so on, contained in a resource dialog box. For example, a client would be a NetWorker resource, and the client's name would be an attribute of that resource. Most resources also have default settings or multiple preconfigured settings to simplify configuration and operation.

The NetWorker product comes packaged with preconfigured and default resource settings. However, to maximize your NetWorker server performance, you can modify these resources to optimize the performance of your particular server(s). Many of the features and commands found in the NetWorker interface, and much of the information contained in this *Administrator's Guide*, have to do with resource creation and configuration.

Table 2 lists NetWorker resources and their descriptions. Some of the resources described in Table 2 also exist as attributes of other resources. These interdependent relationships affect the order in which you implement the configuration, because component resources must be defined before they can be listed as attributes of other resources. NetWorker software explicitly enforces this rule.

Table 2. NetWorker Resources

Resource	Action
Registration	Defines the NetWorker features enabled for this NetWorker server. The Registration resource is described and explained in "Chapter 2: Getting Started" on page 63.
Server	Identifies the local NetWorker server characteristics. Only one server resource can exist per NetWorker server. The Server resource is described and explained in "Chapter 3: Server Management" on page 79.

Reports

The following sections describe various NetWorker reporting functionality you can use.

Routine Data Movement Operations Reports

The degree of success in the completion of scheduled group backups, group cloning, and archive operations is reported to you by the **savegrp** program through a savegroup completion report. This report is the program triggered by the preconfigured notification *Savegroup Completion*. The report is sent as an e-mail to root and to the log file in */nsr/logs/messages*.

The report consolidates the following information:

- The success or failure of each save set participating in the operation
- The operation's save date and time
- The bootstrap *ssid*
- The bootstrap volume location (volume name, starting record number, and ending record number)

A second report, sent to the NetWorker server's designated default printer, repeats the bootstrap information as hard copy, which you should keep on-hand in a secure location. This printed report is the action triggered by the preconfigured notification *Bootstrap*.

Disaster recovery is much easier to perform if you have access to the bootstrap information in the most recent printed report. Refer to the *Disaster Recovery Guide* for further information on using the bootstrap during disaster recovery.

The **nsrinfo** program enables you to query the contents of the NetWorker client file index.

The **nsrwatch** program enables you to use a character-based interface to monitor NetWorker activity as it occurs.

Refer to the *Legato Command Reference Guide* and the man pages for these commands for more information.

Storage Management Application Reports

Table 10 lists the programs that the NetWorker software provides to query the contents of the storage management system. Refer to the *Legato Command Reference Guide* for descriptions of the most commonly used commands and options in more detail.

Table 10. Storage Management Report Programs

Name	Report Generated
mminfo	The contents and mode of the storage volumes and/or the identification numbers and status of the stored save sets.
mmlocate	The user-defined location of storage volumes.
nsrinfo	The contents of the client file index.
nsrmm	The status of the storage devices known to the NetWorker software.

NetWorker Server Statistics and Diagnostic Reports

Messages that report on NetWorker diagnostics are displayed in the NetWorker administrator interface and are also contained in the */nsr/logs/messages* NetWorker messages file. These messages include warning and error conditions and notice of lost connections.

Message Log Files

The messages generated by the NetWorker server daemons (**nsrd**, **nsrindexd**, **nsrmmdbd**, and **nsrmmd**) are contained in the NetWorker *messages* log and the *daemon.log* file, typically found in the */nsr/logs* directory.

Save Set Status Values

The NetWorker server assigns to each backed-up save set a status based on the success of the backup or the age of the save set data. The status of a save set is displayed in the Volumes window. Select Media>Volumes or click the Volumes toolbar button to open the Volumes window. See "How to View Save Set Information" on page 212 for more information.

The save set status changes in the following situations:

- When the save set exceeds its browse policy. For more information about browse policy, see "How the Browse Policy Works" on page 143.
- When the save set exceeds its retention policy, all save sets that are dependent on the save set also exceed their retention policies. For more information about retention policy, see "How the Retention Policy Works" on page 146.
- When you manually change the save set status.

Table 12 provides a list of all the possible values for save set status.

Table 12. Save Set Status Values

Status Value	Meaning	Description
abort	aborted	You aborted the backup for this save set manually or a crash occurred during the operation. This save set is considered immediately eligible for recycling.
brows	browsable	The files in this save set retain entries in the client file index. You can restore all the files using an index-based recover.
inpro	in progress	This save set is currently being backed up.
recov	recoverable	The files in this save set do not have browsable entries in the client file index, and have not passed the retention policy.
recyc	recyclable	The save set and other save sets dependent on this save set for recovery have exceeded their retention policies.

Table 12. Save Set Status Values (Continued)

Status Value	Meaning	Description
scann	scanned-in	The entry for this save set was restored using the **scanner** program. The policies were reset to the values that were applied when the save set was created (if it was created on a 6.0 server) or when the save sets were upgraded to the new 6.0 format (if the save set was created on a pre-6.0 server). For more information, refer to"Scanner Program" on page 279.
susp	suspect	An attempt to recover this save set failed. The **recover** program could not read all the blocks of the save set, for example, if there was a bad spot in the tape.

How to Change the Status of a Save Set

The NetWorker server marks a volume *suspect* if an error occurred while reading the media during backup. You may want to change the status of the save set the server marked as *suspect* to *normal* if you know that the save set data is not really suspect, but the server had difficulty reading the data for other reasons. For example, perhaps the media drive heads were dirty, causing the server to mark the save set *suspect*.

In some cases, you may want to mark a functioning save set as *suspect* in order to omit it from recoveries. This is necessary if you want to recover data from a cloned volume rather than from the original, because the NetWorker server will always look for the original save set during recoveries. Only when the original is *suspect* or non-recoverable will the server then request a cloned save set and its corresponding clone volume. This process requires that a cloned volume containing the cloned save set is available. If a cloned volume containing the cloned save set is not available, then the *suspect* volume is used.

1. Select Media>Volumes to open the Volumes window.
2. Select the Volume whose save sets you want to change in the Volumes attribute.
3. Select Save Set>Change Status>Suspect *or* Normal depending on your needs.

US005535335A

United States Patent [19]

Cox et al.

[11] Patent Number: 5,535,335

[45] Date of Patent: Jul. 9, 1996

[54] METHOD AND SYSTEM FOR REPORTING THE STATUS OF AN AGGREGATE RESOURCE RESIDING IN A NETWORK OF INTERCONNECTED REAL RESOURCES

[75] Inventors: **David E. Cox**, Raleigh; **Karl D. Gottschalk**, Durham; **Craig M. Lawton**; **John F. Linton**, both of Raleigh; **John P. Whitfield**, Apex, all of N.C.

[73] Assignee: **International Business Machines Corporation**, Armonk, N.Y.

[21] Appl. No.: **435,558**

[22] Filed: **May 5, 1995**

Related U.S. Application Data

[63] Continuation of Ser. No. 995,602, Dec. 22, 1992, abandoned.

[51] **Int. Cl.**[6] **G06F 11/34**

[52] **U.S. Cl.** **395/200.11**; 395/839; 395/838; 395/835; 395/837; 395/200.13; 395/600; 364/DIG. 1; 364/282.1; 364/284.4; 364/285.4

[58] **Field of Search** 395/839, 837, 395/835, 838, 200.13, 600; 364/DIG. 1

[56] **References Cited**

U.S. PATENT DOCUMENTS

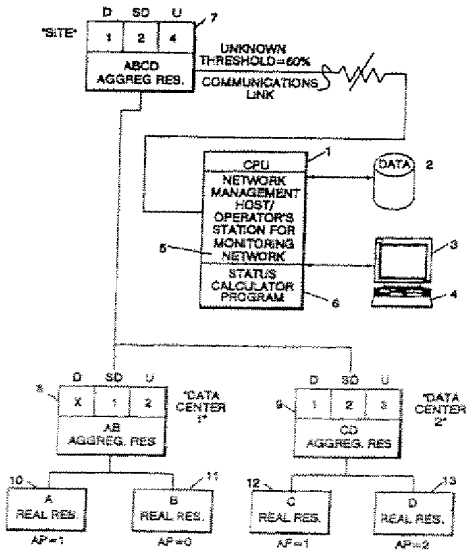
4,769,761	9/1988	Downes	364/514
4,905,171	2/1990	Kiel	364/551.01
5,036,514	7/1991	Downes et al.	364/242.94
5,049,873	9/1991	Robins et al	379/14
5,090,014	2/1992	Polich	371/15.1
5,109,486	4/1992	Seymour	395/200
5,153,837	10/1992	Schaffer	364/464.04
5,181,204	1/1993	Kasman	395/183.21
5,226,120	7/1993	Brown et al.	395/200.11
5,243,543	9/1993	Notess	364/550
5,295,244	3/1994	Dev et al.	395/161
5,299,201	3/1994	Carusone	371/5.1
5,436,909	7/1995	Dev et al.	395/182.02
5,469,463	11/1995	Polich et al.	395/182.18
5,487,148	1/1996	Komori et al.	395/182.02
5,491,791	2/1996	Glowny	395/183.13

Primary Examiner—Thomas G. Black
Assistant Examiner—Jean R. Homere
Attorney, Agent, or Firm—John J. Timar; Edward H. Duffield

[57] **ABSTRACT**

In a network composed of communicating resources, the status of an aggregate resource may be determined by calculations based upon the status assumed by the real resources contained within the aggregate rather than being determined directly. The status of real resources may be propagated to still higher aggregate resources whose own status is based upon the status of the real resources contained within it. An aggregate resource may have multiple potential statuses and a decision as to which status to report for an aggregate resource at a given time is calculated based upon the status of the underlying real resources which it contains. Each real resource has associated with it another parameter which is its aggregation priority value indicating to what degree a change in its status will affect higher level nodes or aggregates in a network hierarchy. In assigning a new aggregate status to a given aggregate resource, the calculation for aggregate status determines the status of the real underlying resources that are descendents of the aggregate resource, retrieves the appropriate threshold levels for status change for the aggregate, and checks the aggregation priority of the real resources that are descendents of the aggregate and are currently in an unsatisfactory state and then assigns a new aggregate resource status.

8 Claims, 4 Drawing Sheets



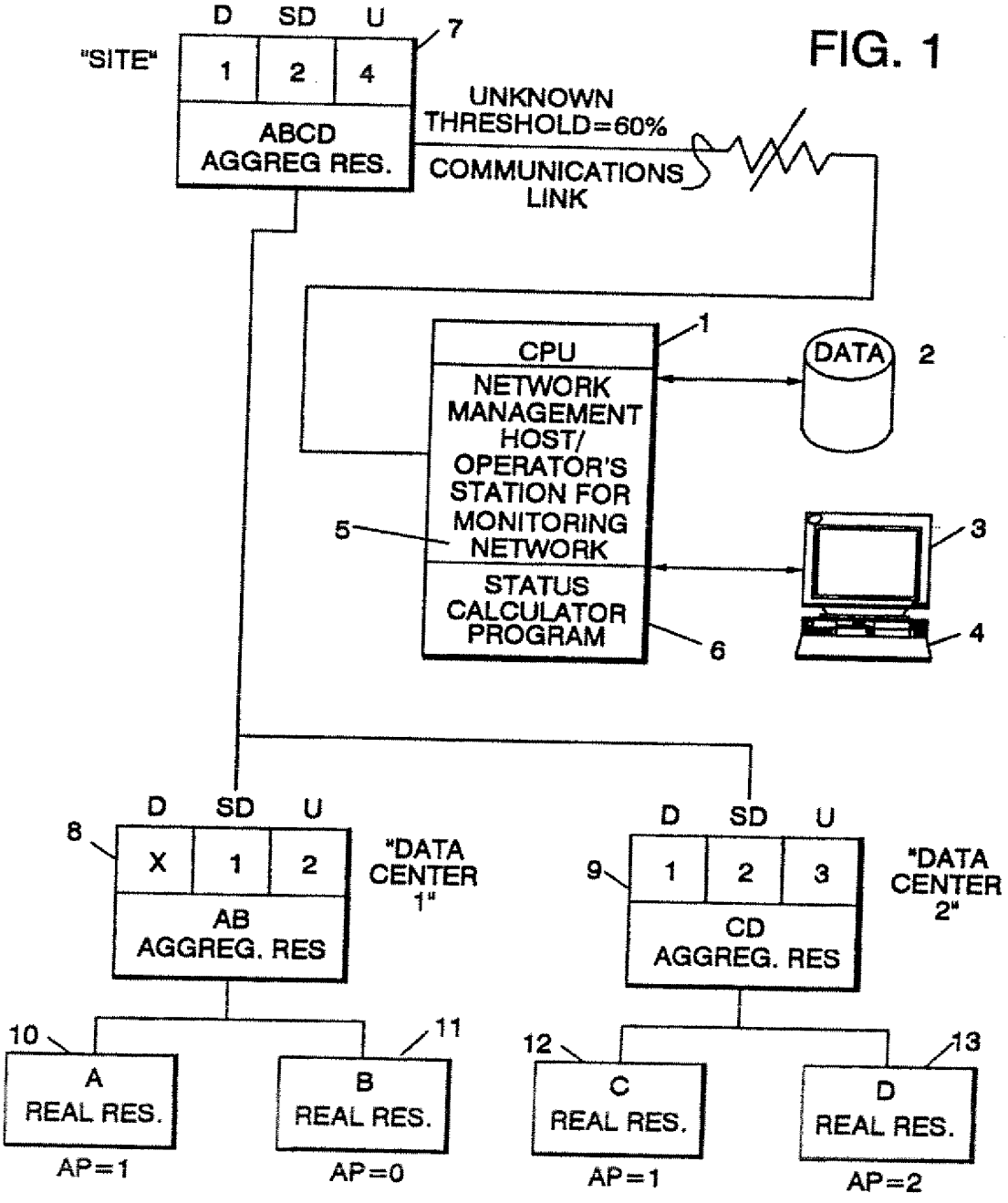
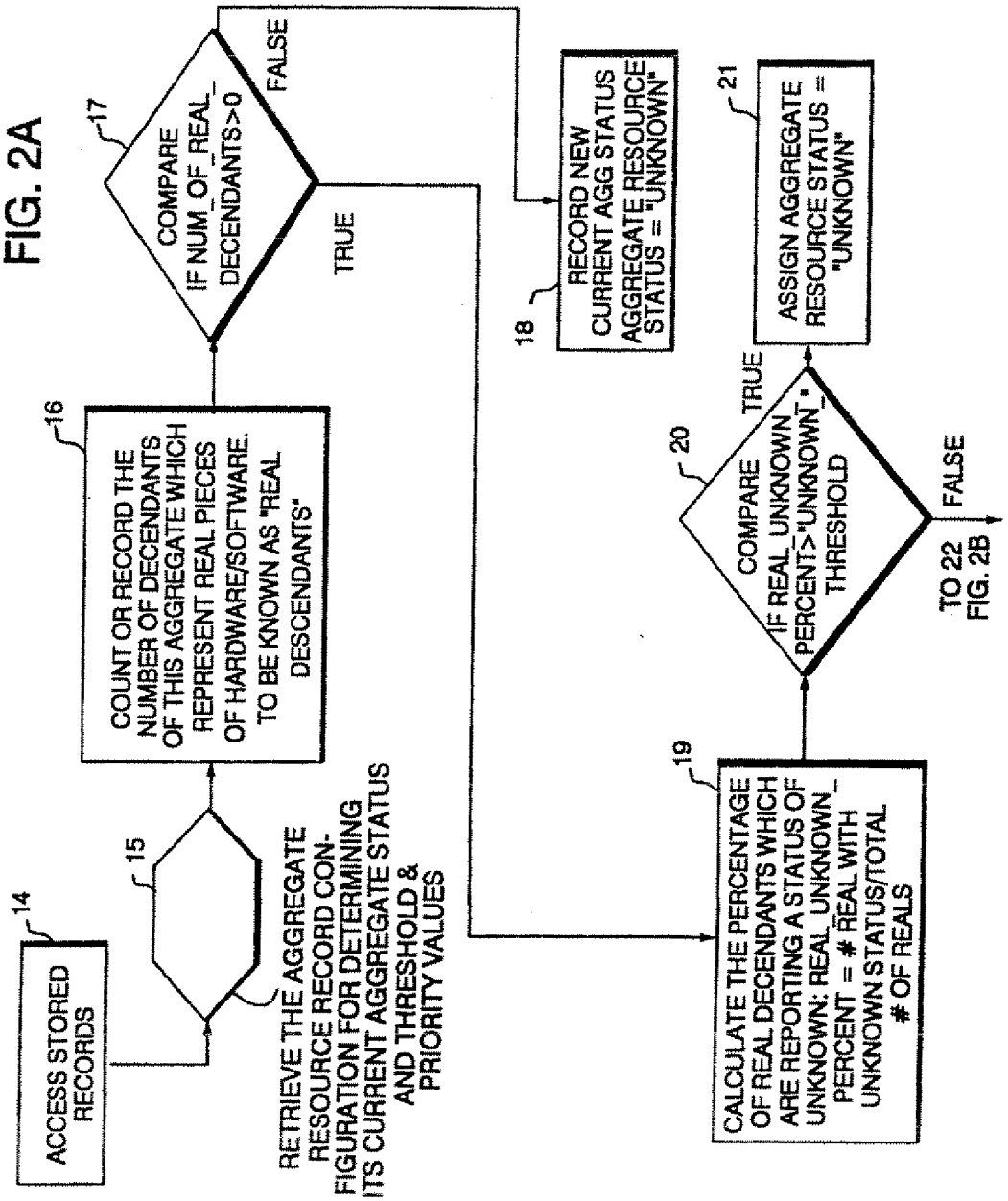

D SD U
"SITE"
1 2 4
7
ABCD AGGREG RES.
UNKNOWN THRESHOLD=60%
COMMUNICATIONS LINK
CPU
1
DATA
2
NETWORK MANAGEMENT HOST/ OPERATOR'S STATION FOR MONITORING NETWORK
5
STATUS CALCULATOR PROGRAM
6
3
4
D SD U
8
X 1 2
AB AGGREG. RES
"DATA CENTER 1"
D SD U
9
1 2 3
CD AGGREG. RES
"DATA CENTER 2"
10
A REAL RES.
AP=1
11
B REAL RES.
AP=0
12
C REAL RES.
AP=1
13
D REAL RES.
AP=2

FIG. 1

FIG. 2A


14
ACCESS STORED RECORDS
15
RETRIEVE THE AGGREGATE RESOURCE RECORD CONFIGURATION FOR DETERMINING ITS CURRENT AGGREGATE STATUS AND THRESHOLD & PRIORITY VALUES
16
COUNT OR RECORD THE NUMBER OF DECENDANTS OF THIS AGGREGATE WHICH REPRESENT REAL PIECES OF HARDWARE/SOFTWARE. TO BE KNOWN AS "REAL DESCENDANTS"
17
COMPARE IF NUM_OF_REAL_ DECENDANTS>0
FALSE
TRUE
18
RECORD NEW CURRENT AGG STATUS AGGREGATE RESOURCE STATUS = "UNKNOWN"
19
CALCULATE THE PERCENTAGE OF REAL DECENDANTS WHICH ARE REPORTING A STATUS OF UNKNOWN; REAL_UNKNOWN_ PERCENT = # REAL WITH UNKNOWN STATUS/TOTAL # OF REALS
20
COMPARE IF REAL_UNKNOWN_ PERCENT>"UNKNOWN_" THRESHOLD
TRUE
FALSE
21
ASSIGN AGGREGATE RESOURCE STATUS = "UNKNOWN"
TO 22 FIG. 2B

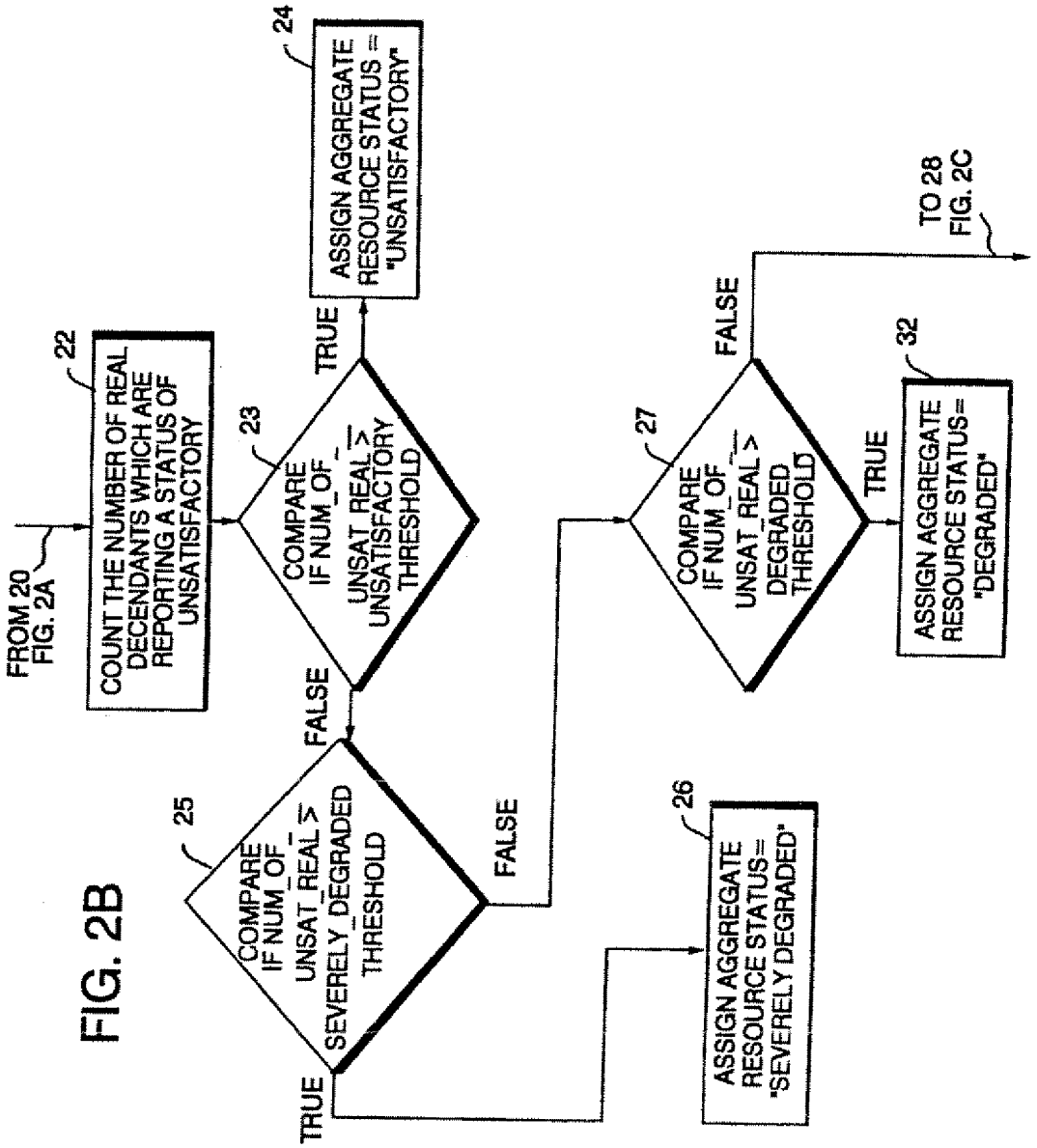
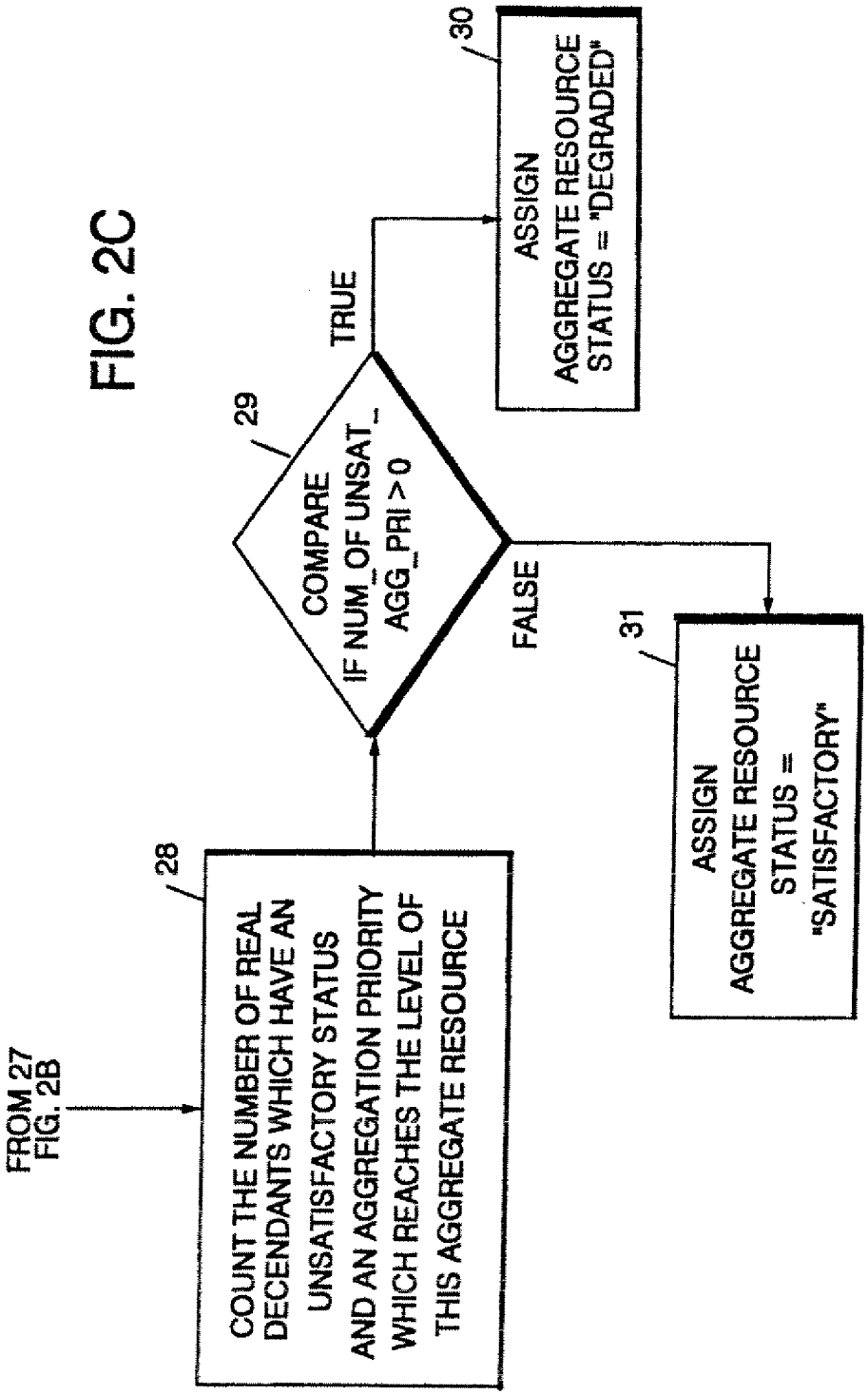

FROM 20 FIG. 2A
22
COUNT THE NUMBER OF REAL DECENDANTS WHICH ARE REPORTING A STATUS OF UNSATISFACTORY
23
COMPARE IF NUM_OF_ UNSAT_REAL > UNSATISFACTORY THRESHOLD
TRUE
24
ASSIGN AGGREGATE RESOURCE STATUS = "UNSATISFACTORY"
FALSE
25
COMPARE IF NUM_OF_ UNSAT_REAL > SEVERELY_DEGRADED THRESHOLD
TRUE
26
ASSIGN AGGREGATE RESOURCE STATUS= "SEVERELY DEGRADED"
FALSE
27
COMPARE IF NUM_OF_ UNSAT_REAL > DEGRADED THRESHOLD
TRUE
32
ASSIGN AGGREGATE RESOURCE STATUS= "DEGRADED"
FALSE
TO 28 FIG. 2C

FIG. 2B

FIG. 2C


FROM 27
FIG. 2B
28
COUNT THE NUMBER OF REAL DECENDANTS WHICH HAVE AN UNSATISFACTORY STATUS AND AN AGGREGATION PRIORITY WHICH REACHES THE LEVEL OF THIS AGGREGATE RESOURCE
29
COMPARE
IF NUM_OF UNSAT_
AGG_PRI > 0
TRUE
30
ASSIGN
AGGREGATE RESOURCE
STATUS = "DEGRADED"
FALSE
31
ASSIGN
AGGREGATE RESOURCE
STATUS =
"SATISFACTORY"

METHOD AND SYSTEM FOR REPORTING THE STATUS OF AN AGGREGATE RESOURCE RESIDING IN A NETWORK OF INTERCONNECTED REAL RESOURCES

This is a continuation of Ser. No. 07/995,602 filed on Dec. 22, 1992, now abandoned.

FIELD OF THE INVENTION

The invention relates generally to interconnected networks of communicating resources composed of either hardware, software or a mix of both typically present in communication network systems. More specifically, the invention deals with a system and method for determining the status of an aggregate resource that is a logical entity containing or composed of one or more real resources which are the hardware or software elements of the network.

PRIOR ART

The problem of maintaining an accurate awareness of the status of logical or aggregate resources in a communications network is an old one. An example of one approach to reporting the status or condition of an aggregate resource such as a LAN network is shown in U.S. Pat. No. 4,769,761 commonly assigned to the assignee of the present application. In this prior patent, stations of an aggregate communications network such as a logical ring or LAN each generate and transmit error reports containing error counts to a central ring error monitor at one of the stations. The error monitor analyzes the report and calculates and stores weighted error counts which are integrated over a selected time interval and compared with a threshold value found normal or acceptable for a communications network. Exceeding the threshold value sets an error flag to indicate possible station failures. This approach provides a rough estimation of the potential viability of the aggregate resource to continue operation but does not provide any means of calculating the actual status based upon the conditions of the underlying real elements contained within the logical aggregate and gives only a gross indication of the potential operative or inoperative status, not a real assessment of actual condition.

An assessment of the actual operative status of an aggregate resource at any given time must reflect both the nature of the aggregate resource and the degree of inoperability being experienced by one or more of the contained real resources which make up the aggregate.

OBJECTS OF THE INVENTION

In light of the foregoing known difficulties with the prior art, it is an object of this invention to provide an improved method and a system for determining the aggregate status based upon the actual status conditions of the contained underlying real resources in the network.

BRIEF SUMMARY

The foregoing and still other objects of the invention which are not specifically enumerated are met in a preferred embodiment described herein in which a calculation process and system are described for determining the aggregate resource status based on the status of the underlying real network resources. A real resource is defined as an actual operative piece of hardware or software residing within a network, for example a communications network arranged

in some hierarchical or other interconnected topological arrangement whose interconnection scheme and last reported physical status are maintained in a network topological and status database by a network management system such as the IBM Corporation's NetView product. Such network management products are arranged to receive reports generated by network elements and management facilities whenever the state of a monitored resource changes. An aggregate resource is defined as a logical resource composed of or containing a number of real resources. Its status is calculated based upon the status of the real resources that it contains rather than being determined by its own characteristic since a determination based upon the real status of underlying elements will prove more reliable in estimating the ability of the aggregate resource to function. In the preferred embodiment, each aggregate resource has associated with it an aggregate status generated or determined based upon the status of the underlying real resources that it represents. If multiple levels of aggregate resource exist, each may in turn contain multiple real resources and the aggregate status of a high level aggregate resource is based upon the status of all real resources that it contains directly or indirectly by containing lower level aggregate resources which in turn may contain still further real resources.

In the preferred embodiment, the network resource topology and interconnection status database is queried to determine which real resources are contained within a given aggregate resource or hierarchy of aggregate resources, and what the status of each real resource is. Stored parameter values are used as threshold indicators for determining when the status of the aggregate resource should be modified. Each aggregate resource will have associated with it in its status database several parameter values which are prestored by a system operator based upon his or her understanding of the connectivity and real resource personality contained in the system.

A first parameter value indicates a threshold number or value for underlying real resource elements that must be found in an unsatisfactory state before the status of the aggregate resource is reflected as being degraded. The second parameter value indicates a number of underlying real resource elements that must be found in an unsatisfactory state before the system determines that the status of an aggregate resource should be changed from degraded to severely degraded and a third parameter value represents a threshold for changing the aggregate resources status from severely degraded to unsatisfactory.

The status determination calculation system and method perform aggregate status calculation determinations based upon the status of the real resources that are monitored which are found to be descendents of the aggregate rather than on the status of any child aggregates contained within the higher level aggregate itself. A fourth parameter is a stored threshold value associated with each real resource describing its aggregation priority and indicating to the determination system and method how much effect a change in status of a contained real resource should have upon a higher level parent or grandparent aggregate. This threshold value is a number from 0 to n where n is the number of layers in the network and indicates how far up the hierarchical tree of the network to turn the parent or grandparent of an aggregate of the real resource being monitored to a status of degraded. In assigning a new status for an aggregate resource, the calculation method and system look at the status of the real resources that are the descendents of the aggregate resource, determining an aggregate status for the

aggregate resource based upon the threshold parameter values associated therewith and also upon the aggregation priority of any affected real resources which are descendents of the aggregate and currently residing in an unsatisfactory state.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing objects are met in a preferred embodiment of the invention that well be described in greater detail with reference to the drawings in which:

FIG. **1** illustrates a high level schematic topological arrangement of a network of resources including real resources and aggregate resources whose status is to be determined.

FIGS. **2A–2C** illustrate a typical flowchart for the process carried out by the system in the preferred embodiment for determining the status of an aggregate resource based upon the reported status conditions of underlying real resources, parametric values obtained from a database and comparison conditions.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENT

FIG. **1** schematically illustrates an interconnected network that may represent communication elements or other entities in a layered or hierarchical arrangement with respect to one another. The example shown in FIG. **1** may be taken as an illustration of a typical computer data processing network having the CPU, data storage, network management programs, a host operator station for monitoring the network with a display, a keyboard, etc. connected via a communications link to an aggregate resource generally termed a site or locus which may include a plurality of subservient computers which might be data centers each of which may contain one or more real resources such as work stations or subservient computers or other data generating devices.

The aggregate resources such as the data centers or the site may be known to the network management host operator only as an aggregate without real detail of the underlying, real resources that make up the highest level "site" aggregate. Of course, access to a repository of data containing the topological interconnection scheme and details of the physical and logical entities may be used to generate an overall view or picture of the aggregate resource as is commonly done with network management products such as the IBM Corporation's NetView program. However, merely knowing the configuration or identity of the elements making up the network and their interconnection scheme does not directly lead to an understanding of what the present operative status of an aggregate resource may be since only the real physical or software resources actually report conditions of their operation to their network management system at the host.

Each real resource may assume several possible status conditions such as: "satisfactory", indicating that most or all of its capability is in satisfactory working order, or "unsatisfactory", indicating that most or all or a preponderance of the real resources are in an unsatisfactory status. Other statuses or types of status might be envisioned, but for the purposes of this illustration these are sufficient.

The status of an aggregate resource is not as easily determined since it may contain multiple real resources both hardware and software which may have divergent status and in which a minor degradation of one real resource among several contained within an aggregate may not significantly affect the operability status of the aggregate as a whole.

In order to address the problem of providing an accurate assessment of the operability status of an aggregate resource, the present invention makes use of the network topology and status database to contain a plurality of threshold parameters used for assessing the condition of real resources contained within an aggregate resource and in forming computational and analytical assessments of the effect of their status upon the overall status of the aggregate resource.

Turning to FIG. **1**, the overall configuration and topology of an interconnected network for communications or reporting or other hierarchical or management system of resources is shown. In FIG. **1**, the host CPU **1** comprises a database **2**, display **3**, an operators input station **4** and a network management program **5** which gathers information on the operative status and condition present in the network from all of the known reporting real resource elements. A status calculator program **6** resides in host **1** for execution at the direction of the operator at work station **4**. A communications link connects the CPU to the entities that are represented by an aggregate resource **7**, termed a "site" or installation. The installation contains two additional aggregate resources termed "data center **1**" and "data center **2**" which are numbered **8** and **9**, respectively. Each data center **1** and **2**, as an aggregate resource, also contains real physical resources of hardware or software nature numbered **10–13** as shown.

Each aggregate resource has several parametric values for it stored in database **2** by the operator or the network management program **5** at initiation of operation or updated by human intervention. Parameters indicating the levels at which degraded, severely degraded or unsatisfactory performance of an aggregate resource are to be determined are stored as threshold values in the database **2**. For convenience, these values are indicated as register or counter contents and are depicted in the aggregate resources **7**, **8** and **9** as being present as values for those aggregate resources, although the values need not be physically stored at the location where the aggregate resources are defined and would normally be stored in the database **2**.

An additional parametric value is contained in database **2** for the threshold of "unknown" condition set at an arbitrary value which will be described in greater detail later. Similarly, each real resource has stored for it in the database a threshold parameter for its "aggregation priority" of effect on higher level parent, grandparent, etc. as will be discussed in greater detail later.

A status calculator program **6** operates in the host CPU **1** under control of the network management code **5** and the host operator via the control station **4**. These are used to determine the aggregate status for an aggregate resource utilizing program means operating in accordance with a method that will be described in greater detail.

FIG. **1** illustrates an arbitrary example, not necessarily of a communications environment, in which there are a number of elements in a network connectivity hierarchy in which the status of the aggregate resources **7**, **8** and **9** is to be monitored by a monitoring CPU and management program. Four real resources A, B, C and D are illustrated as residing at the resource locations identified by the numerals **10**, **11**, **12** and **13**, respectively. Their status will be the determinative factor in deciding the status of the aggregate resources. The aggregate resources contain or are superior to the real resources according to the example given in FIG. **1**, so that the data center **1** which is identified with the numeral **8** is an aggregate resource owning or representing the real resources A and B, while aggregate resource **9** represents resources C

and D that are real resources. The higher level aggregate resource **7** comprises the aggregate resources **8** and **9** so that it includes the real resources A, B, C and D within its environment.

Let us suppose that all four real resources **10–13** are initially in a satisfactory state and that resource A (numeral **10**) goes into an unsatisfactory or failure state. The status calculator program and method **6** in the CPU **1**, when called upon to determine the status of the aggregate resource **7** or **8**, will first determine the number of real resources that are the descendants of the aggregate resource **8**, the parent resource for resource A. If the aggregate resource **8** does not have any real resources as descendants, its status will be set to "unknown" and the calculation will be complete. In this example, real descendants of aggregate resource **8** do exist, so the status calculator **6** will determine the ratio of real descendants having a status of "unknown" to the total number of real descendants, since in this example neither real resource A or B has a status value of "unknown", the ratio will be 0. The ratio is compared in the status calculator **6** to a threshold ratio for determining a status of "unknown" for the aggregate resource AB. In the example illustrated in FIG. **1**, this threshold value is taken as 60%. If the calculated ratio turns out to be greater than or equal to this threshold value, the status of the aggregate resource AB will be determined by the status calculator **6** to be "unknown" and the calculation will end. However, in this example, the ratio that is calculated is less than the threshold for attaining "unknown" status, so the calculation continues.

Next the status calculator will examine parameter values stored in association with the aggregate resource **8**. The first value is the parameter value indicating the number of real resources that must be found in an unsatisfactory state in order to determine that the aggregate resource **8** should also be in a "degraded" state. In the example illustrated in FIG. **1**, the value of this parameter is listed under the "D" heading as being contained in a register, counter or memory cell in association with the aggregate resource **8**, although it would be understood that the actual parameter value would be stored in the database **2** for examination by the status calculator program **6**. The first parameter value is set to X in this example, indicating that the aggregate resource **8** will not be put into a degraded state under any circumstances except by the aggregation priority, described on page 10. This is an arbitrary example and instead of X, a value of 1 or 2, etc. might have been entered.

The second parameter value is the content of the counter or register memory space identified by "SD" for "severely degraded". This is the threshold value at which the aggregate resource **8** should be placed in the severely degraded state. In this example, this parameter value is set to 1, indicating that the aggregate resource **8** will be placed in the severely degraded state if at least one underlying real resource A or B is found to be in an unsatisfactory or failed state.

The third parameter value is identified by "U" and stands for the threshold level at which the aggregate resource **8** will be put into the "unsatisfactory" state. The content of this parameter is **2**, indicating that the aggregate resource **8** will be put in the unsatisfactory state when at least two underlying real resources have been detected to be in an unsatisfactory state.

In the example outlined, the information contained in the parameters is compared with the example that a single real resource, resource A, has entered the unsatisfactory state and the aggregate resource **8** will be tentatively assigned a status of "severely degraded" since 1 is equal to or greater than the threshold level 1 for the aggregate resource **8** entering into the severely degraded state.

There is, however, a sensitivity in networks of this type in which an underlying real resource failing or entering an unsatisfactory state may be so serious in its effect that the entire aggregate resource should also be placed in a degraded state. To handle this possibility based upon the nature of each real resource and its potential effects upon higher levels in the network, the status calculator **6** will look at still a fourth parameter value assigned significance as an aggregation priority threshold for each real resource independently. The aggregation priorities are arbitrarily assigned based on the nature of each real resource and are shown in FIG. **1** as AP=1 next to real resource **10**, AP=0 next to real resource B, etc. For the assumed example, real resource **10** A has failed or reached an unsatisfactory state. Its aggregation priority is equal to 1 indicating that the aggregate status of an aggregate resource one level higher than the real resource should be assigned a degraded status. However, the status calculator program **6** has already tentatively assigned a status of "severely degraded" for the aggregate resource **8** based on reaching of the severely degraded threshold parameter value. The status calculator will therefore select the more serious state of degradation, "severely degraded", as the calculated status of the aggregate resource **8**.

Next the status calculator **6** will be asked to calculate or determine the status of the aggregate resource **7** containing the aggregate resources **8** and **9** which in turn contain the real resources A, B, C and D. The aggregate resource **7** has four real "grandchildren" as descendants, all of which have a known status so the calculation of the ratio of the number of real descendants having "unknown" status to the total number of real descendants will yield 0. Therefore the status calculator **6** will look at the values of the threshold parameters contained in the database **2** for each of the real resources contained by the aggregate resource **7** by way of aggregate resources **8** and **9**. The status calculator will tentatively assign the aggregate resource **7** a status of "degraded" since the value of the parameter D for aggregate resource **7** is a 1 and one underlying real resource, resource A, has been found to have an unsatisfactory status. The aggregation priority of resource A is 1, also indicating that only the direct parent, i.e., the aggregate resource **8** for the real resource **10** will be affected by a change in status of the real resource. Therefore, the indicated status of the aggregate resource **7** will be set to "degraded" based upon the threshold value comparison with the calculated or reported status conditions of the underlying real resources.

A flowchart of the process steps carried out by the status calculator program **6** in CPU **1** is given in FIG. **2**. The first step is to access the stored records in the database as shown in the beginning block **14** of the flowchart in FIG. **2**. In block **14**, the calculator program **6** retrieves the aggregate resource topology configuration and status records to determine the current aggregate status shown for an aggregate resource and to retrieve the threshold parameters and aggregation priority parameter values as shown in block **15**. Next the program examines the retrieved record and determines the number of real descendants which a given aggregate resource represents as shown in block **16**. In block **17**, the program **6** compares the number of real descendants found for the aggregate resource in question to 0. If there are no real descendants, block **18** is entered and the aggregate resource status is set to "unknown". However, if real resources do exist, block **19** is entered and the program calculates ratio of the number of real resources having a status of unknown to the total number of real resources

which are descendants of the aggregate resource being studied.

In block **20** the calculated ratio or percentage is compared against a threshold value retrieved in blocks **14** and **15**. If the number of real descendants reporting an unknown status exceeds an arbitrary threshold value (60% as shown in FIG. **1**) the comparison in block **20** will direct the flow to block **21** so that the aggregate resource being studied will be assigned a status of "unknown". However, if the ratio is less than the threshold ratio parameter value in block **20**, block **22** will be entered and program **6** will determine the number of real descendants for the aggregate resource that are currently reporting a status of unsatisfactory.

In block **23** the number of real descendants of the aggregate resource that report a status of unsatisfactory are compared to a second parameter value representing the threshold at which the aggregate resource is to be set into the unsatisfactory status. If the comparison indicates the number of real resources is greater than or equal to this parameter value, block **24** is entered and the aggregate resource is assigned a status of unsatisfactory. However, if the comparison in block **23** is false, block **25** is entered where a further comparison is made between a third parameter value, indicating the level at which a severely degraded threshold exists for the aggregate resource, and the number of real descendant resources of the aggregate resource being studied that have been detected as being in an unsatisfactory condition. If the comparison shows the number of real resources having an unsatisfactory condition is greater than or equal to this third threshold parameter value, block **26** is entered and the aggregate resource is assigned a status of severely degraded. However, if the comparison in block **25** is false, block **27** is entered.

In block **27** a fourth comparison is made against another threshold parameter value obtained in blocks **14** and **15**. The number of real resource descendants to the aggregate resource being studied which are reporting an unsatisfactory condition are compared with this fourth threshold parameter value. If the number of real, unsatisfactory status resources is greater than or equal to this fourth threshold, block **32** is entered and the aggregate resource is assigned a status of degraded. However, if the comparison in block **27** is false, block **28** is entered. The number of real descendants of the aggregate resource under study which have an unsatisfactory status and an aggregate priority parameter value which is equal to the level in the hierarchical network occupied by the particular aggregate resource under study is determined. Then block **29** is entered where the number of real resources having an unsatisfactory condition and an aggregation priority of greater than or equal to the level of the particular aggregate resource is found. If any of the real resources in an unsatisfactory condition have an aggregation priority that is greater than or equal to the level of the particular aggregate resource, block **30** is entered and the aggregate resource status is set to degraded, but if none of these real resources has an aggregation priority that is greater than or equal to the level in the heirarchical network occupied by the particular aggregate under study, block **31** is entered and the aggregate resource status is calculated to be satisfactory.

In the examples of the preferred embodiment just described, it should be apparent that the initial parameter values and priority values that may be set into the database **2** for each resource are arbitrary and based upon the initial assumed default values. The user or system operator can easily update or change the priority or parameter values. Typically, tables of parameter values that correspond with the various types of resources are made available with a network management resource program such as **5** shown in FIG. **1**. The manufacturer assigns parameter values based upon the significance of a failure or status change of a given element of a certain type in a network or hierarchical arrangement. As new types of resources are made available to the users, new tables of values can easily be shipped for updating or augmenting the database, but no changes to the process and program **6** will be required since it will generically act on the values obtained from the data tables for the individual resource types. It will go through the same calculation and comparison steps without changing the logic of the status calculation for aggregate resources.

Therefore, what is desired to be protected by letters patent is set forth in the following claims by way of example only and not of limitation in which what is claimed is:

1. A computer implemented method for determining the operational status of an aggregate resource residing in a network of interconnected real resources, said aggregate resource being defined as a logical resource including a plurality of real resources and represented by a configuration record stored in a database wherein said configuration record contains a list of said real resources and a current operational status and an aggregation priority value for each said real resource and further contains a plurality of predefined parameter values that are used to determine the operational status of said aggregate resource, said method being executed by a monitoring computer and comprising the steps of:

determining from said configuration record the number of real resources contained within said aggregate resource and having an operational status of a first type;

comparing said number of real resources having the operational status of said first type with said plurality of predefined parameter values and assigning said aggregate resource an operational status corresponding to the largest of said predefined parameter values that is equalled or exceeded by said number of real resources having the operational status of said first type; and

if said number of real resources having the operational status of said first type is less than the smallest of said predefined parameter values, determining if any of said real resources having the operational status of said first type also has a non-zero aggregation priority value, such that:

for each said real resource having the operational status of said first type and said non-zero aggregation priority value, determining if the operational status of said real resource should be assigned as said operational status of said aggregate resource;

if the operational status of any of each said real resource should be assigned as the operational status of said aggregate resource, assigning said operational status of said real resource to said operational status of said aggregate resource; and

if the status of each said real resource should not be reflected in said operational status of said aggregate resource, assigning said aggregate resource an operational status of satisfactory.

2. The method as claimed in claim **1** further comprising the steps of:

determining from said configuration record the number of said real resources contained within said aggregate resource and the number of said real resources having the operational status of a second type;

calculating a ratio of the number of said real resources having the operational status of said second type to the

number of real resources contained within said aggregate resource;

retrieving a threshold value from said configuration record and comparing said threshold value to said calculated ratio; and

if said calculated ratio exceeds said threshold value, assigning the operational status of said second type to said aggregate resource.

3. The method as claimed in claim 1 wherein said aggregate resource includes a plurality of other aggregate resources with each of said other aggregate resources including a plurality of real resources, said step of determining if the operational status of said real resource should be assigned as said operational status of said aggregate resource includes:

arranging said plurality of other aggregate resources and said plurality of real resources in a hierarchical structure and assigning a sequential level number to each level in said hierarchical structure, said aggregate resource being assigned the highest sequential level number;

propagating said operational status of said real resource upwards a number of levels in said hierarchical structure corresponding to said aggregation priority value of said real resource.

4. The method as claimed in either of claim 1 or 2 further comprising the step of recording said operational status of said aggregate resource in said configuration record.

5. A computer system for determining the operational status of an aggregate resource residing in a network of interconnected real resources, said aggregate resource being defined as a logical resource including a plurality of real resources and represented by a configuration record stored in a database wherein said configuration record contains a list of said real resources and a current operational status and an aggregation priority value for each said real resource and further contains a plurality of predefined parameter values that are used to determine the operational status of said aggregate resource, said system comprising:

means for determining from said configuration record the number of real resources contained within said aggregate resource and having an operational status of a first type;

means for comparing said number of real resources having the operational status of said first type with said plurality of predefined parameter values and assigning said aggregate resource an operational status corresponding to the largest of said predefined parameter values that is equalled or exceeded by said number of real resources having the operational status of said first type; and,

if said number of real resources having the operational status of said first type is less than the smallest of said predefined parameter values:

means for determining if any of said real resources having the operational status of said first type also have a non-zero aggregation priority value;

for each said real resource having the operational status of said first type and said non-zero aggregation priority value, means for determining if the operational status of said real resource should be assigned as said operational status of said aggregate resource;

means for assigning said operational status of said real resource to said operational status of said aggregate resource if the operational status of any of each said real resource should be assigned as the operational status of said aggregate resource; and,

means for assigning said aggregate resource an operational status of satisfactory if the status of each said real resource should not be reflected in said operational status of said aggregate resource.

6. The system as claimed in claim 5 further comprising:

means for determining from said configuration record the number of said real resources contained within said aggregate resource and the number of said real resources having the operational status of a second type;

means for calculating a ratio of the number of said real resources having the operational status of said second type to the number of real resources contained within said aggregate resource;

means for retrieving a threshold value from said configuration record and comparing said threshold value to said calculated ratio; and,

means for assigning the operational status of said second type to said aggregate resource if said calculated ratio exceeds said threshold value.

7. The system as claimed in claim 5 wherein said aggregate resource includes a plurality of other aggregate resources with each of said other aggregate resources including a plurality of real resources, said means for determining if the operational status of said real resource should be assigned as said operational status of said aggregate resource including:

means for arranging said plurality of other aggregate resources and said plurality of real resources in a hierarchical structure and assigning a sequential level number to each level in said hierarchical structure, said aggregate resource being assigned the highest sequential level number; and,

means for propagating said operational status of said real resource upwards a number of levels in said hierarchical structure corresponding to said aggregation priority value of said real resource.

8. The system as claimed in either of claims 5 or 6 further comprising means for recording said operational status of said aggregate resource in said configuration record.

* * * * *